

Présentation générale du BE tir laser

1. Description du jeu de tir à réaliser

Le jeu se compose de 6 pistolets et d'une cible. Voici une description rapide :

- **6 pistolets laser.** Lors de l'appui sur la gâchette, un rayon laser est émis durant 100ms (ne jamais regarder directement le faisceau, voir les consignes de sécurité inscrites sur chaque pistolet). Le rayon contient l'identifiant du pistolet. On ne peut tirer que deux « cartouches » par seconde. Les pistolets sont mis à disposition. Aucun travail n'est à faire dessus.

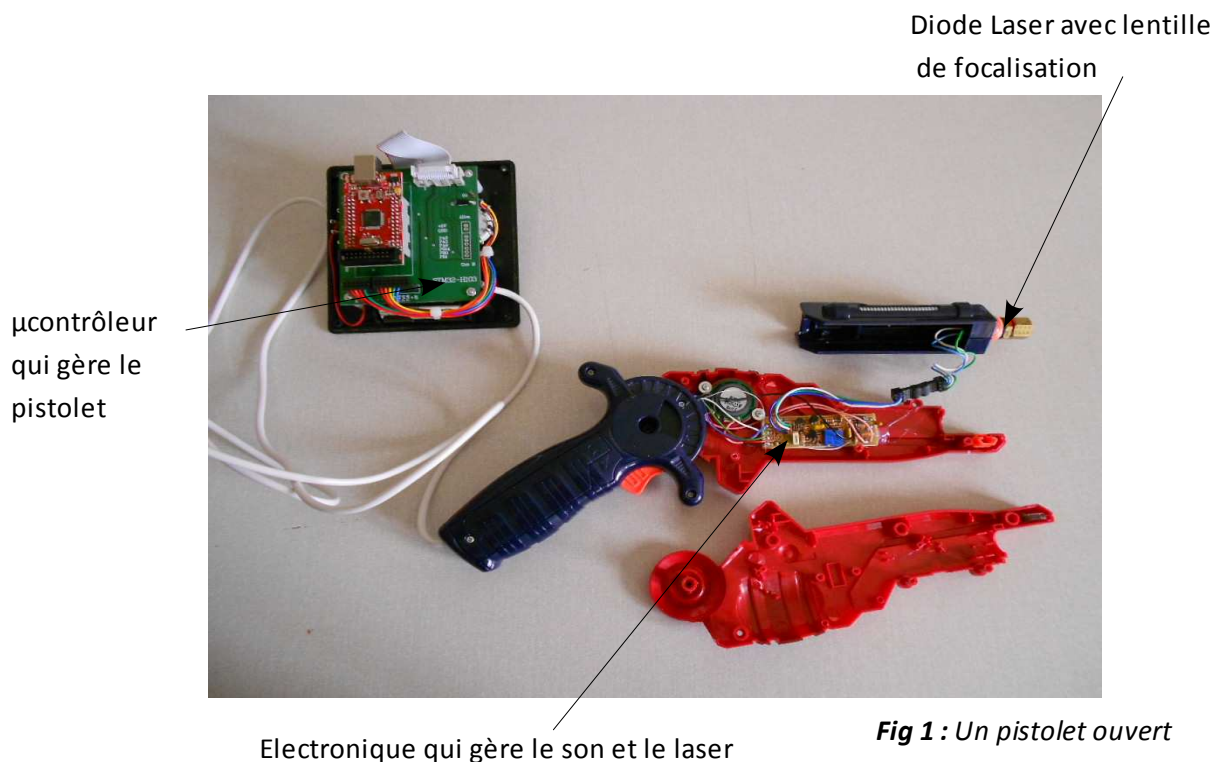


Fig 1 : Un pistolet ouvert

– **Une cible** est composée de :

- | | | |
|----------------|---|---|
| façade | { | – 4 capteurs photosensibles (1 seul utilisé à la fois), |
| | | – deux LED de visualisation par capteur, par exemple pour indiquer les moments où les tirs sont autorisés, |
| | | – un haut parleur qui émet un son typique, par exemple bris de verre, |
| | | – un affichage de score (non visible sur les photos car en cours de fabrication...), |
| Gestion du jeu | { | – un montage électronique pour traiter le signal récupéré par le capteur , |
| | | – un microcontrôleur de type STM32F103RB (Cortex M3) qui traite l'ensemble des éléments de la cible. C'est lui qui gère le jeu. |



Fig 2 : La façade de la cible

Les règles du jeu sont décrites ci-après. Elles seront entièrement (ou presque) programmées en assembleur. La planification des TP est faite de manière à ce que tout étudiant passe forcément par des étapes clé (gestion d'un son, programmation d'une DFT). Selon l'efficacité de travail de chacun, plusieurs règles du jeu pourront être élaborées.

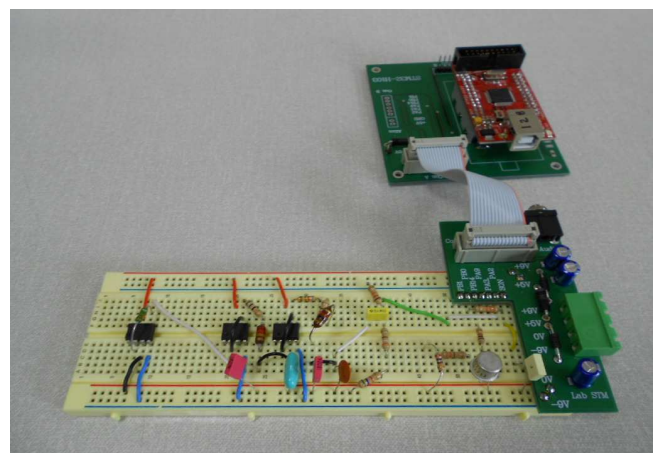


Fig 3 : Réalisation sur plaque faite par un binôme :
L'électronique et le micro-contrôleur associés à la cible

2. Règles du jeu

Règle du jeu de base :

Chacun des six joueurs vise la cible. Chaque fois qu'un joueur fait mouche, un son est produit. Le score de chacun des joueurs est géré et mis à jour à chaque impact.

Option A : Les tireurs ne peuvent agir que dans une fenêtre de temps donnée. Les LED rouges et vertes seront utilisées pour donner l'information.

Option B : Le son est géré au plus près de la réalité, c'est-à-dire qu'un impact se traduit immédiatement par un son, y compris si un son précédent est en cours. Il faut donc gérer plusieurs émissions sonores simultanément. Un impact = un son (pouvant durer une seconde).

Option x : Faites votre proposition...

Remarque :

La règle du jeu de base est à faire complètement en assembleur. Dès lors qu'une option est envisagée, l'application peut être réalisée en langage C. La grande majorité du travail fait en assembleur sera ré-exploité. Le logiciel sera donc un mélange de C et d'assembleur.

3. Principe physique mis en oeuvre au travers du jeu

La question principale qu'on peut se poser, est de savoir par quel moyen la cible va-t-elle pouvoir identifier chacun des pistolets. Parmi toutes les réponses possibles, nous avons retenu l'utilisation d'une analyse spectrale, autrement dit, la mise en oeuvre d'une transformée de Fourier discrète. Chaque pistolet émettra donc pendant 100ms un signal périodique dont la fréquence est parfaitement connue et stable. De son côté, la cible procèdera périodiquement à des analyses de Fourier pour identifier le tireur, gérer le score, émettre un son.

4. Présentation rapide du travail à faire sur 13 séances de TP

Les 13 séances sont très guidées. Seules deux sont dédiées à une amélioration personnelle du jeu (règle du jeu plus évoluée). Nous avons donc fait le choix d'une progression naturelle dans laquelle tous les étudiants développent les mêmes parties du jeu. Par ailleurs, il y a dans ce BE des concepts nouveaux qui méritent une formation particulière, sorte de parenthèses dans le projet. Ces apports de connaissances se feront sys-

tématiquement sous la forme d'un support *pdf* ou papier et d'une application pratique. Il n'y a pas de cours classique dans ce BE.

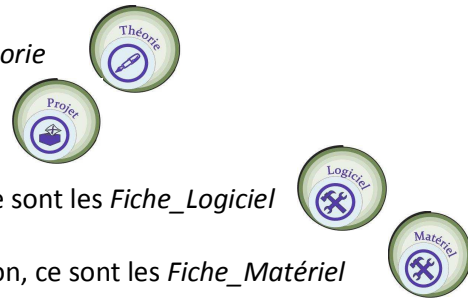
Les enseignants qui vous encadrent en TP n'ont pas à faire de cours préparé au tableau. Leur rôle est simplement de répondre aux questions que vous vous posez à la lecture des documents. Les enseignants sont là pour vous aider à surmonter une difficulté. Il est donc **absolument nécessaire de lire le cours avant de venir en TP** et d'être actif, c'est-à-dire de solliciter les enseignants.

4.1. Les documents, les supports pdf et autres sources

Tous les documents et téléchargements divers sont disponibles sur *Moodle* (PO MIC 3ème année / Bureau d'étude d'intégration 2).

Les documents sont classés en 4 types :

- les apports théoriques nouveaux, ce sont les *Fiche_Theorie*
- la description du travail à faire, ce sont les *Fiche_Projet*
- les guides et autres manuels portant sur les logiciels, ce sont les *Fiche_Logiciel*
- les documents qui décrivent le matériel mis à disposition, ce sont les *Fiche_Matériel*



4.2. La progression pédagogique

Séance 1 et 2 : Etude de la transformée de Fourier discrète (Discrete Fourier Transform, DFT). Vous devrez vous approprier le cours associé. Pour comprendre les subtilités de cette opération, vous utiliserez *Matlab*, conjointement avec *Simulink*. Vous comprendrez alors comment régler les paramètres de la DFT ainsi que ses limitations.

Séances 3 à 5 : L'électronique de traitement du capteur optique. Vous réaliserez un filtre passe-bas d'ordre supérieur à 1 (plus complexe qu'un simple RC) ainsi que deux autres fonctions indispensables. Vous apprendrez à câbler, mettre au point, dépanner votre montage. Vous consoliderez vos compétences en matière de mesure (notamment sur l'oscilloscope). A la fin de cette étape, vous disposerez de l'ensemble de l'électronique nécessaire à l'application.

Contrôle de connaissance, la DFT essentiellement + électronique

Séance 6 à 11: Durant ces six séances, vous allez développer votre code embarqué sur le micro-contrôleur STM32F103RB. Le langage est l'assembleur dont vous aurez au préalable acquis les bases (cours et TD assembleur).

La progression est guidée, voici les étapes :

- | | |
|------------------------|---|
| <i>Produire un son</i> | <ul style="list-style-type: none"> – création d'une petite temporisation logicielle (boucle for...), – exploitation de la temporisation logicielle pour jouer un son (gestion de tableau), – produire un son avec une temporisation <u>matérielle</u> (utilisation d'une fonction d'interruption), |
| <i>Vers la DFT</i> | <ul style="list-style-type: none"> – Vérification par calcul en format fractionnaire de la relation trigonométrique bien connue :
 $\cos^2(\theta) + \sin^2(\theta) = 1,$ – programmation d'une DFT 8 points, – programmation d'une DFT 64 points en lien direct avec l'application, |
| <i>Projet : le jeu</i> | <ul style="list-style-type: none"> – Intégration des fonctions assembleur pour élaborer le jeu (règle du jeu de base). |

Séance 12 et 13: Autonomie complète, poursuite du projet en cours, amélioration de la règle du jeu, migration possible vers un logiciel mixte C/ASM.

Présentation orale 20 mn environ par binôme, note individualisée.

5. Les outils nouveaux de ce BE

Simulink : Environnement graphique associé à MATLAB. Utile pour la partie traitement de signal, DFT.

KEIL : assembleur, débogage en simulation, débogage in situ (sur cible réelle), (mélange C / Assembleur)

6. Méthode générale

- *Découpez votre travail : Un programme complexe doit être découpé en plusieurs portions de code simples que l'on testera au fur et à mesure. Valable pour du logiciel comme pour du matériel (un montage lourd se découpe en plusieurs de petite taille).*
- *Anticipez vos tests : Lorsque la phase de conception est achevée, que vous vous lancez dans la réalisation (codage ou câblage), vous devez avant toute chose savoir comment vous allez vous y prendre pour valider ce que vous vous apprêtez à réaliser.*